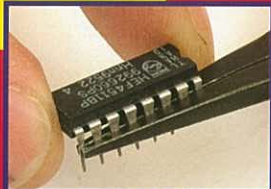
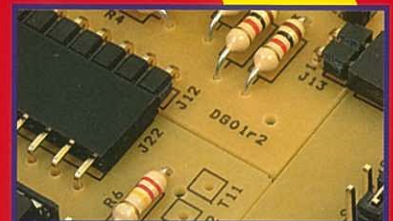
...e costruisci il tuo **LABORATORIO DIGITALE**

6,90 €

11

A	B	C	f(A,B,C)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1

DIGITALE DI BASE

HARDWARE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41011

Peruzzo & C.

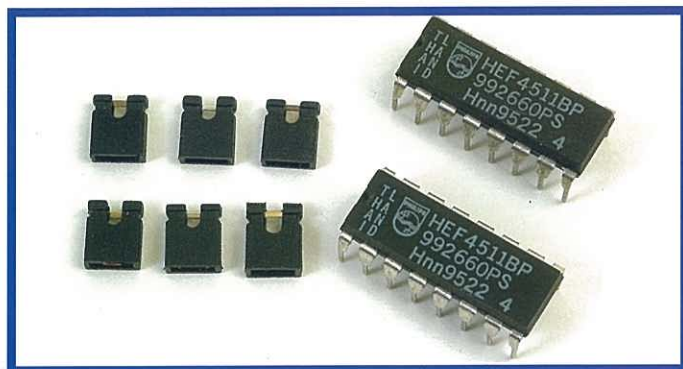
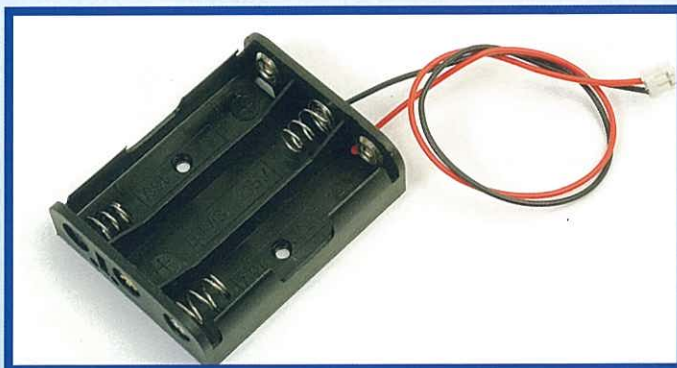
Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Staroffset s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE" si compone di 70 fascicoli settimanali da suddividere in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

IN REGALO in questo fascicolo

2 Circuiti integrati 4511
6 Ponticelli isolati

IN REGALO nel prossimo fascicolo

1 Portabatterie con connettore femmina a 2 poli

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

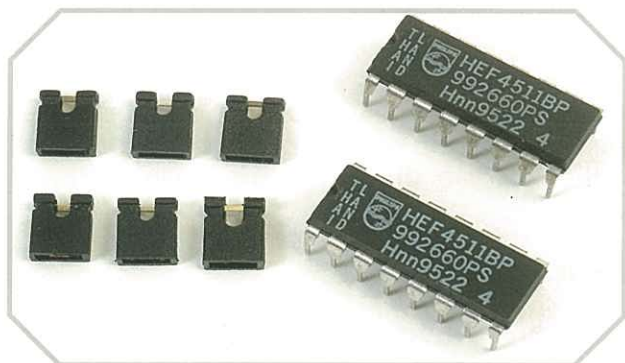
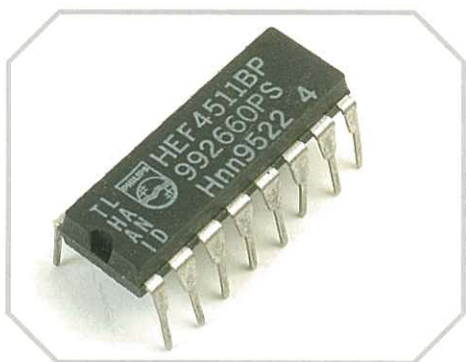
L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede. **Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:** elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

Installazione del driver

Materiali di questo numero.

In alcuni casi i terminali si piegano durante il trasporto.

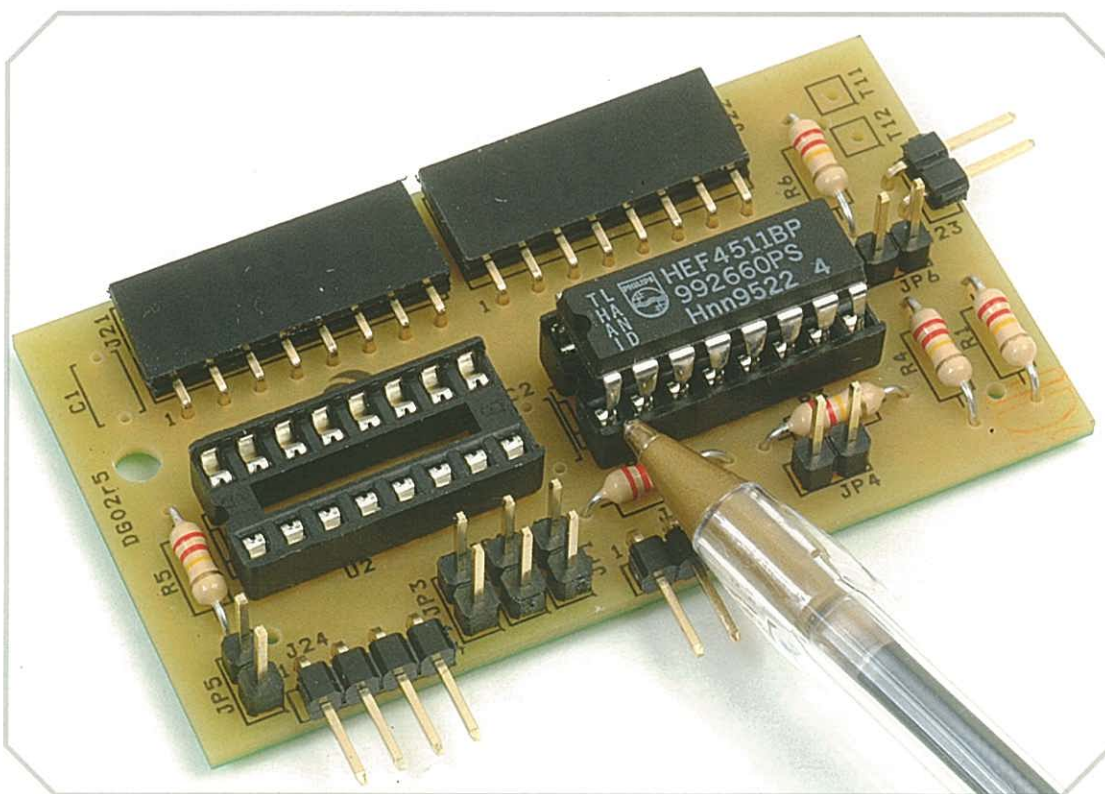
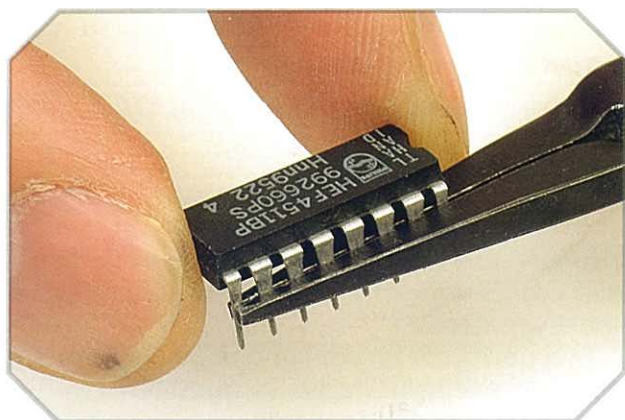
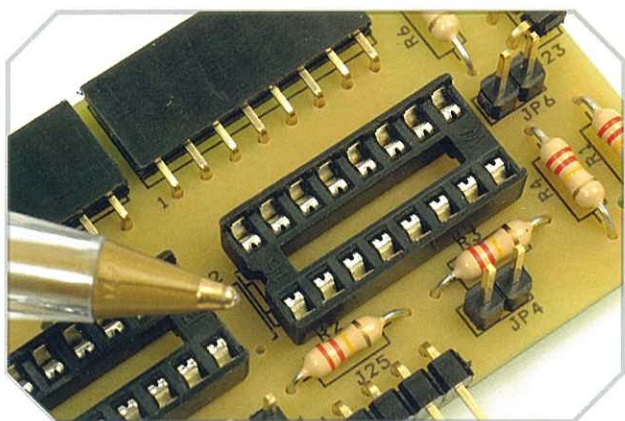
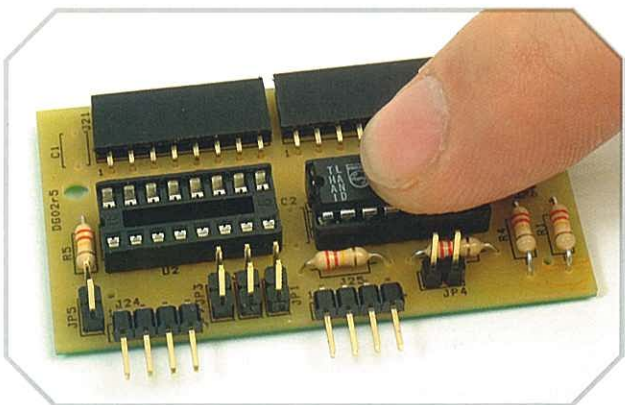
In questo fascicolo vi viene fornito il materiale necessario per rendere operativa la scheda DG02, che corrisponde ai driver per i display a sette segmenti.

I due circuiti integrati driver 4511 della famiglia CMOS si montano inserendoli nei loro rispettivi zoccoli.

I sei ponticelli invece sono di uso generale e possono essere utilizzati in questo circuito o su altri per poter realizzare i diversi esperimenti.

Circuiti integrati

I circuiti integrati si montano nei rispettivi zoccoli della scheda. I piedini di questi circuiti vengono inseriti direttamente negli zoccoli. Per alcuni lettori questa operazione risulterà molto semplice e in realtà lo è, tuttavia occorre comunque adottare alcune precauzioni per evitare problemi.

Per fare in modo che l'inserzione sia effettiva ed efficace tutti i piedini dell'integrato de-

Prima di stringere i piedini dell'integrato per la sua inserzione ne dobbiamo verificare l'orientamento, identificando il terminale 1.

I terminali devono essere raddrizzati per facilitare l'inserzione.

Le tacche di riferimento indicano come deve essere inserito l'integrato.

Bisogna stringere leggermente i piedini dell'integrato per assicurare il contatto dei suoi terminali.

vono essere perfettamente allineati con i corrispondenti dello zoccolo, inoltre devono essere perfettamente integri non solo per un buon fissaggio dell'integrato ma anche per permettere alle piccole molle di connessione dello zoccolo di fornire un buon contatto elettrico.

Manipolazione

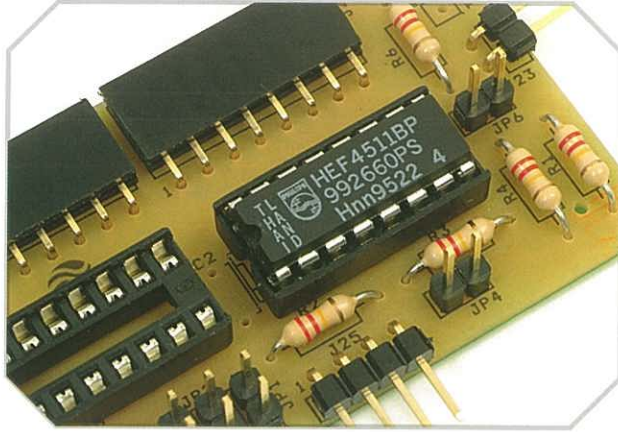
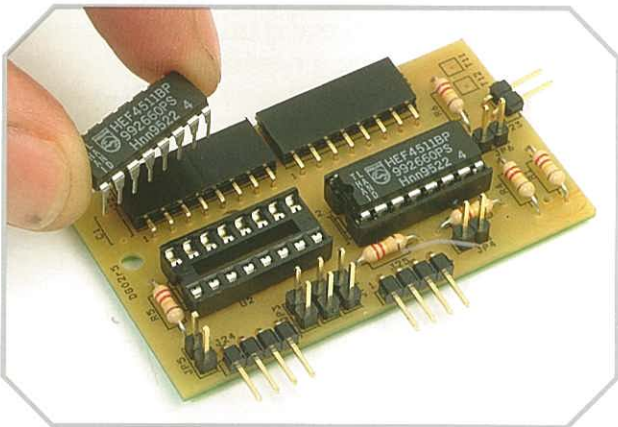
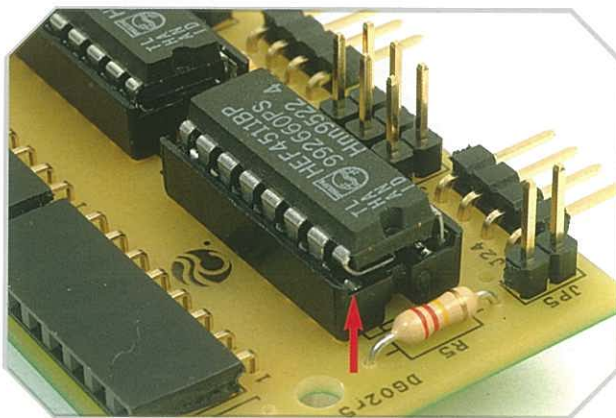
In generale bisogna evitare di toccare con le dita i terminali dei circuiti integrati prendendoli ai due lati. Anche se i moderni circuiti integrati CMOS sono ben protetti contro le scariche elettrostatiche, a volte questa protezione è insufficiente, in quanto le scariche possono essere molto forti e danneggiare i circuiti interni.

Nei laboratori sono normalmente previste molte prese di terra e zone per potersi scaricare toccando delle superfici conduttrici collegate a massa, inoltre solitamente si usano delle calzature con suola conduttrice e indumenti antistatici. Anche il pavimento del laboratorio deve avere determinate caratteristiche evitando vernici o cere che compromettano le sue proprietà di conduzione. Quando si lavora a casa un buon consiglio è quello di toccare con la mano la zona dove è posto l'integrato prima di toccare l'integrato stesso e, se è possibile, scaricarsi su qualche zona metallica, anche se è piuttosto difficile sapere se ha un buon collegamento a massa.

Orientamento

Il circuito integrato ha due file di piedini quindi esiste il rischio di collegarlo al contrario invertendo la fila. Per evitare questo errore, viene inciso il lato dell'integrato posto tra il primo e l'ultimo piedino con una tacca di riferimento che ritroveremo poi sia sullo zoccolo che sulla serigrafia della scheda; queste tacche di riferimento sono realizzate con diversi metodi e sono comunque sempre presenti.

Su alcuni integrati, in sostituzione della tacca, vi è un punto vicino al terminale 1. La serigrafia può essere fatta in diversi modi ma è sempre facile identificare il terminale 1.

È interessante notare, invertendo la scheda, che una delle piazzole di stagno dove verranno saldati i terminali è diversa, nel nostro caso è quadrata, e questo permette di identificare visivamente in modo rapido e sicuro il termi-

Circuito con un integrato montato.

Montaggio del secondo integrato.

Questo pin si è piegato perché non era ben allineato.

nale 1 del circuito integrato e utilizzarlo come riferimento per identificare gli altri. Questo particolare è molto utile al momento di eseguire misure con puntali o sonde di misura.

Terminali

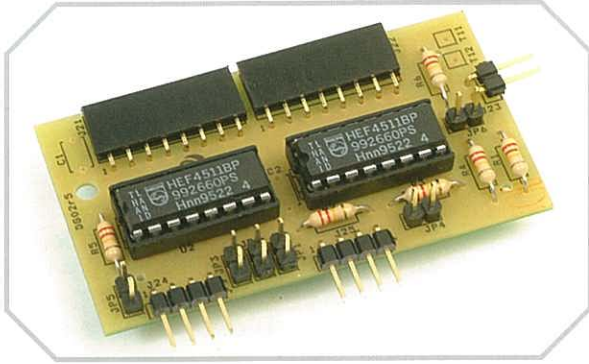
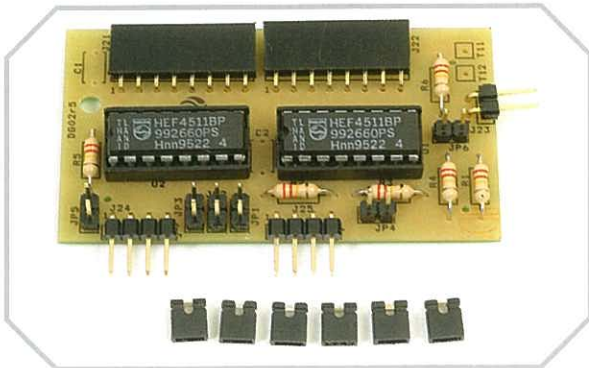
Nonostante vengano prese tutte le precauzioni per cercare di danneggiare il meno possibile i terminali degli integrati durante il trasporto, in alcuni casi si potrebbero determinare delle deformazioni degli stessi, che generalmente non ne influenzano il funzionamento. Quindi prima di inserire un circuito integrato è necessario controllare e, se è il caso, raddrizzare i suoi terminali per fare in modo che entrambe le file siano perfettamente allineate e corrispondano ai terminali dello zoccolo.

Se non si prende questa precauzione è possibile che qualcuno dei terminali si pieghi verso l'esterno facendo pressione sull'integrato; questo lo si rileva facilmente, però potrebbe succedere che si arrotoli o che si pieghi verso l'interno, passando inosservato. In tutti questi casi il contatto con il terminale corrispondente dello zoccolo sarebbe inadeguato, causando un malfunzionamento. Un altro problema che si potrebbe presentare si produce quando, nonostante i terminali siano perfettamente allineati, non fanno un buon contatto; questo potrebbe succedere perché i terminali dell'integrato non sono stati inseriti a sufficienza, ovvero non è stata fatta sufficiente pressione al momento di inserire il componente nello zoccolo.

Ponticelli

I ponticelli forniti possono essere utilizzati su questa scheda o su qualsiasi altra che ne abbia bisogno. Sulla scheda DG02 si possono usare su JP5 come TEST di illuminazione dei segmenti, o su JP6 per forzare lo spegnimento di tutti i segmenti. Gli altri ponticelli si possono inserire in JP1, JP2, JP3 e JP4 per formare un codice di ingresso sul driver U1. L'assenza di ponticelli si interpreta come un 1 logico e la loro presenza come uno 0 logico.

Smontaggio delle schede

Nel caso sia necessario smontare la scheda DG02 bisognerà svitare ed estrarre la vite di fis-

Ecco come deve risultare il circuito con i componenti forniti.

I ponticelli permettono di fare prove con la scheda.

saggio e ripetere la stessa operazione anche per la vite di DG01. Conviene anche allentare, ma non togliere, le due viti della scheda DG04.

Le schede DG01 e DG02 si tolgono contemporaneamente dal lato esterno, tirandole entrambe. Dopo averle tolte si possono separare l'una dall'altra.

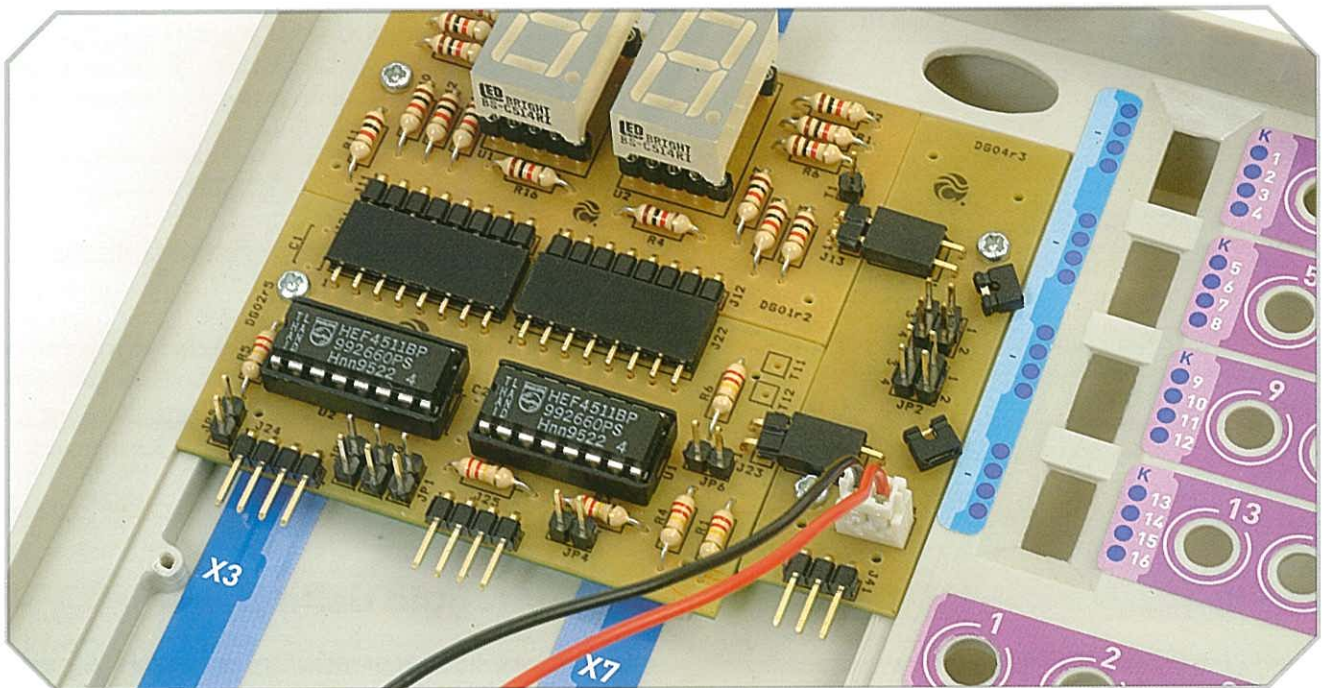
Per montarle nuovamente bisogna seguire il procedimento inverso, ovvero assemblarle tra loro prima di collegarle alla scheda di alimentazione DG04.

Piccoli componenti

Quasi tutto il materiale che vi viene fornito, specialmente quello di maggiori dimensioni, dovrà essere montato sul laboratorio.

Oltre a questo vi verrà fornito un altro tipo di materiale, ad esempio i ponticelli e una serie di componenti sciolti che si utilizzeranno nelle differenti prove e che è necessario conservare.

Conviene procurarsi una piccola scatola per contenere questi componenti in modo che non si perdano e tenerli sempre a portata di mano per quando sarà necessario.

Ecco come dovrà essere il circuito. Se le pile sono inserite toglieremo i ponticelli dalla scheda DG04 per scollegare l'alimentazione dai circuiti DG01 e DG02.

Funzioni e tabelle della verità

Le tabelle della verità non si utilizzano solamente per rappresentare la funzione logica di una determinata porta di un circuito, servono anche per rappresentare circuiti complessi o, in altre parole, funzioni più complesse di quelle che corrispondono a una semplice porta logica.

Tabelle della verità

Anche se abbiamo già parlato delle tabelle della verità, ricordiamole nuovamente dato che sono realmente uno strumento utile per studiare le funzioni algebriche.

Per prima cosa studieremo la relazione che esiste tra una tabella della verità e una funzione, più avanti vedremo come si ottiene un circuito che corrisponda a questa funzione.

Una tabella della verità è formata da file (orizzontali) e colonne (verticali).

La prima cosa da analizzare è il numero di variabili della funzione, perché avremo bisogno di tante colonne quante sono le variabili.

Il numero di file dipende dal numero di variabili e si calcola come 2 elevato al numero di variabili. In altre parole per due variabili sono necessarie quattro file e il numero si duplica per ogni variabile aggiunta. Ad esempio, per tre variabili sono necessarie otto file.

Il risultato, ovvero la funzione, si rappresenta su un'altra colonna.

A	B	C	f(A,B,C)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Esempio di tabella della verità di una funzione.

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Tabella ausiliaria X.

A	B	C	/A	Y
0	0	0	1	0
0	0	1	1	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

Tabella ausiliaria Y.

Anche una funzione può essere rappresentata mediante differenti espressioni, queste devono essere equivalenti, perché una funzione ha una tabella della verità unica.

Funzioni

Illustriamo come si può rappresentare una funzione con una tabella della verità attraverso un esempio. Data la funzione:

f(A, B, C) = ABC + /AB + /C(A + B)

scomporremo ognuno degli elementi della funzione OR:

ABC = X
/AB = Y
/C(A + B) = Z

e rappresenteremo ognuno su una tabella della verità, utilizzando tabelle intermedie ausiliarie per evitare errori :

Tabella ausiliaria Z.

A	B	C	/C	(A+B)	Z
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	0	1	0
1	1	0	1	1	1
1	1	1	0	1	0

Una volta ottenuti gli operandi ausiliari applicheremo la funzione OR a questi tre elementi e otterremo la tabella della verità della funzione:

Tabella della verità della funzione.

A	B	C	X	Y	Z	X+Y+Z	f(A,B,C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	1	1	1	1
0	1	1	0	1	0	1	1
1	0	0	0	0	1	1	1
1	0	1	0	0	0	0	0
1	1	0	0	0	1	1	1
1	1	1	1	0	0	1	1

Utilizzo delle proprietà

Una funzione può avere diverse espressioni. Applicando le proprietà logiche è possibile cambiare facilmente l'espressione, ricordando però che questa rappresenta sempre un'unica funzione. Possiamo utilizzare lo stesso esempio di prima:

f(A, B, C) = ABC + /AB + /C(A + B) =
ABC + /AB + /CA + /CB =
ABC + (/A + /C)B + /CA =
BAC + (/A + /C)B + /CA =
B/(/A + /C) + (/A + /C)B + /CA =

Utilizziamo ora un'espressione ausiliaria:

/A + /C = M
B/M + MB + /CA =
B(/M + M) + /CA =
B(1) + /CA =
B + /CA

A titolo di verifica, calcoleremo la tabella della verità di questa nuova espressione, osservando che si arriva alla stessa tabella della verità, il che dimostra che questa nuova espressione è corretta, dato che rappresenta la stessa funzione.

A	B	C	/C	/CA	f(A,B,C)
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0

Tabella della verità della funzione, con un'altra espressione.

Proprietà

Le tabelle della verità sono anche molto utili per verificare alcune proprietà.

A	B	A+B	/(A+B)	/A	/B	/A/B
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Teorema di Morgan.

A	/A	A+/A
0	1	1
1	0	1

Esperimenti con codice binario

Disponiamo già di tre circuiti stampati con componenti sufficienti per realizzare alcuni esperimenti. Sono i circuiti stampati DG01, DG02 e DG04. Abbiamo anche un portabatterie che può alimentare l'insieme formato dai tre circuiti precedenti dopo che saranno stati assemblati tra loro. Questa alimentazione è possibile perché è stato previsto un connettore ausiliario J44 sulla scheda del circuito DG04, dove per il momento possiamo collegare il connettore di uscita del portabatterie.

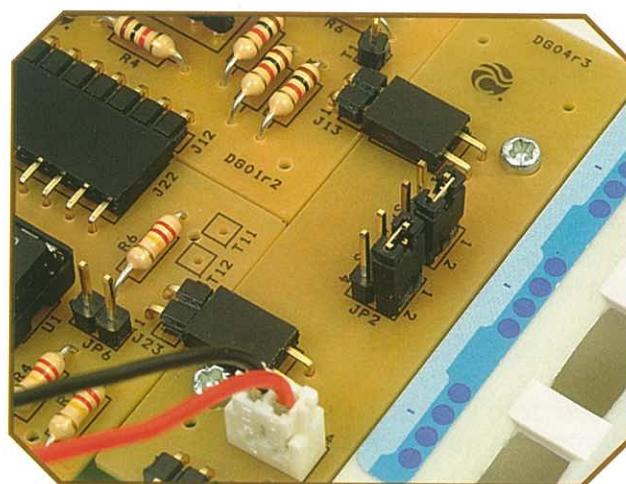
L'esperimento

Questo esperimento è facile da realizzare, dato che ha solamente bisogno di essere preparato. Consiste nell'applicare un codice a 4 bit all'ingresso di uno dei circuiti integrati driver 4511 che formano il circuito DG02, in modo che il risultato si visualizzi sul display delle unità.

Questo esperimento può essere molto utile per fare pratica con il codice binario, dato che il display ci indicherà se abbiamo sbagliato o meno ad applicare il codice a 4 bit. Verificheremo anche l'utilizzo dei terminali ausiliari del driver LT e BI.

Materiali

I materiali necessari sono i circuiti DG01, DG02 e DG04, con i componenti che vi abbiamo for-

Una delle prime cose da fare è verificare che le pile siano montate esattamente.

Montando i ponticelli 1 e 2 di JP1 e JP2 della scheda DG04, forniremo alimentazione ai circuiti DG01 e DG02.

nito finora già montati, in ogni caso quando le schede saranno completate l'esperimento si potrà ripetere senza alcun tipo di problema. Oltre a questi circuiti è necessario il portabatterie, tre pile da 1,5 volt tipo AA o R6 e i sei ponticelli a due terminali che vi sono stati forniti.

Preparazione

I tre circuiti devono essere collegati tra loro, inoltre è consigliabile avvitarli al laboratorio.

All'inizio della prova nessun ponticello deve essere montato né sul circuito di alimentazione DG04, né sul driver DG02, tuttavia i circuiti devono disporre, come minimo, dei componenti che possiamo vedere nelle immagini e devono essere state realizzate le saldature secondo quanto indicato nella sezione hardware.

Alimentazione

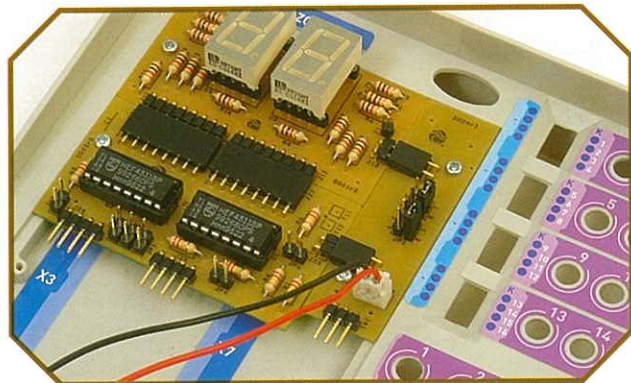
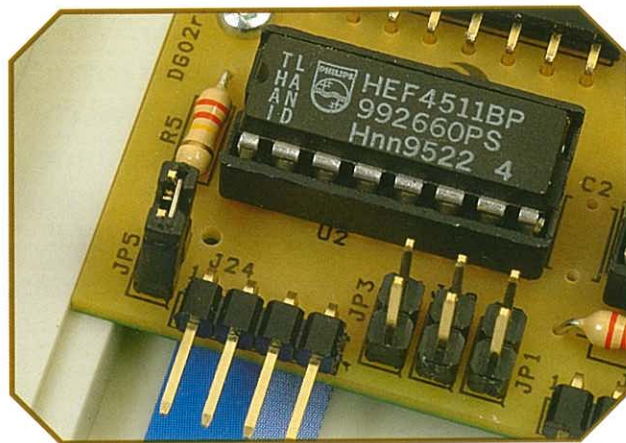
I circuiti DG01 e DG02 si alimentano tramite il circuito DG04, benché esista la possibilità di alimentarli a 5 e a 9 volt quando il laboratorio sarà più completo, nel cui caso occorrerà montare i ponticelli JP1 e JP2 sui terminali 3 e 4 per alimentare a 9 volt, e sui terminali 1 e 2 per alimentare a 5 volt; quest'ultima posizione si utilizzerà anche per il connettore J44 del portabatterie, però in questo caso si alimenterà solamente a 4,5 volt, sufficienti per alimentare questo tipo di integrati e per realizzare l'esperimento.

Quando il laboratorio sarà completo potremo ripetere l'esperimento a 9 V, ottenendo più luminosità sui segmenti del display.

Installando le pile bisognerà fare attenzione alla loro posizione e non ci dovremo dimenticare di montare i ponticelli che permetteranno al circuito di prendere energia dalle batterie.

Codice binario

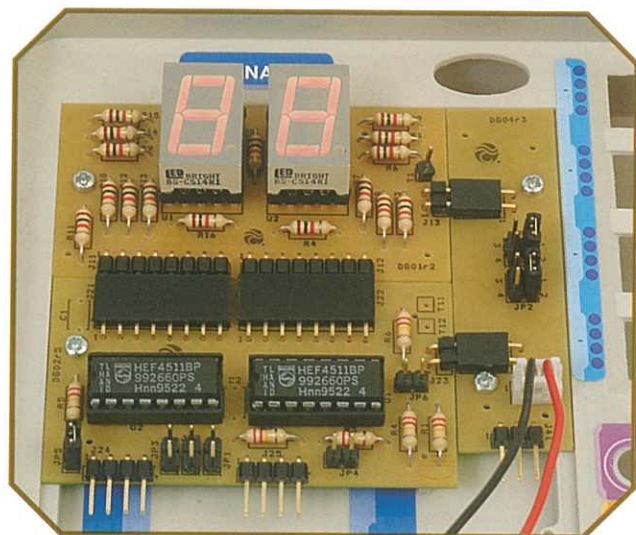
Rivediamo il codice binario e la sua corrispondenza con quello decimale. In questa prova si utilizzeranno i ponticelli JP1, JP2, JP3 e JP4 che insieme alle resistenze R1, R2, R3 e R4 permetteranno di selezionare il codice binario applicato ai quattro ingressi del circuito integrato U1, un driver 4511 della famiglia CMOS, come possiamo vedere nello schema della scheda Digitale Avanzato 19.

I quattro ingressi sono simili, quando non si collega il ponticello la resistenza da 220 K stabilisce un livello logico alto all'ingresso a cui è

All'inizio della prova il circuito DG02 deve essere nella sua condizione normale, ovvero non deve avere nessun ponticello inserito.

Il ponticello per i TEST è JP5.

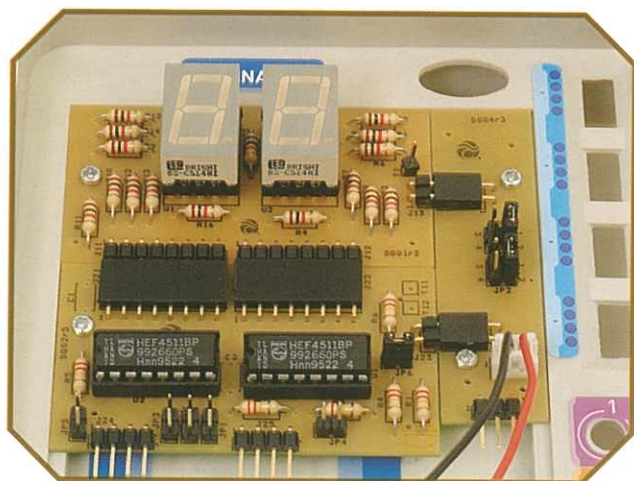
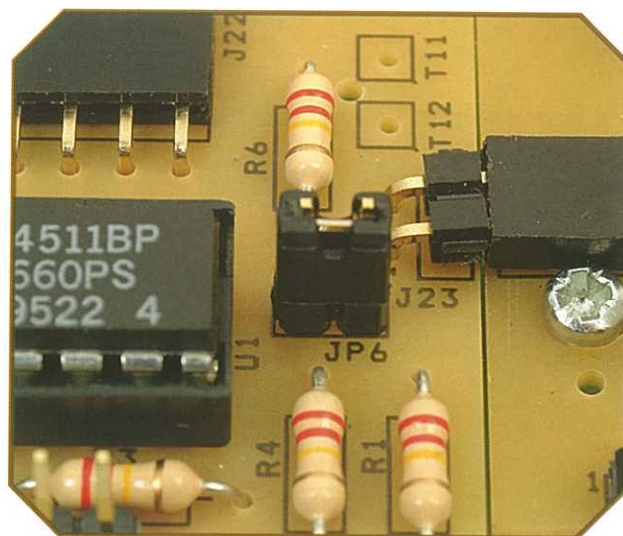
collegata e questo si interpreta come 1, inserendo però il ponticello questo ingresso passa a livello logico basso. Quando il ponticello è collegato si rappresenta con ON (0 logico), e quando non lo è con OFF (1 logico). Vediamo la tabella riassuntiva:

D C B A	JP4	JP3	JP2	JP1	Decimale	Display
0 0 0 0	ON	ON	ON	ON	0	0
0 0 0 1	ON	ON	ON	OFF	1	1
0 0 1 0	ON	ON	OFF	ON	2	2
0 0 1 1	ON	ON	OFF	OFF	3	3
0 1 0 0	ON	OFF	ON	ON	4	4
0 1 0 1	ON	OFF	ON	OFF	5	5
0 1 1 0	ON	OFF	OFF	ON	6	6
0 1 1 1	ON	OFF	OFF	OFF	7	7
1 0 0 0	OFF	ON	ON	ON	8	8
1 0 0 1	OFF	ON	ON	OFF	9	9
1 0 1 0	OFF	ON	OFF	ON	10	spento
1 0 1 1	OFF	ON	OFF	OFF	11	spento
1 1 0 0	OFF	OFF	ON	ON	12	spento
1 1 0 1	OFF	OFF	ON	OFF	13	spento
1 1 1 0	OFF	OFF	OFF	ON	14	spento
1 1 1 1	OFF	OFF	OFF	OFF	15	spento
Nota: ON ponticello inserito, OFF senza ponticello						

Test dei segmenti

Dopo aver montato i ponticelli tra 1 e 2 di JP1 e JP2 del circuito DG04, le schede riceveranno alimentazione. Se osserviamo i display vedremo che uno dei due, quello delle unità, rimane completamente spento, mentre l'altro po-

Inserendo il ponticello su JP5 si illuminano tutti i segmenti.

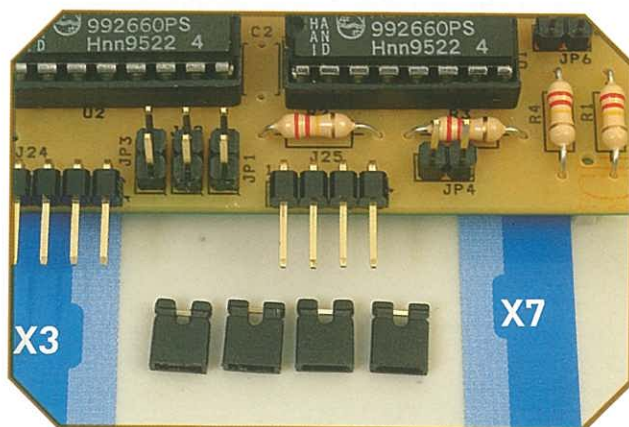
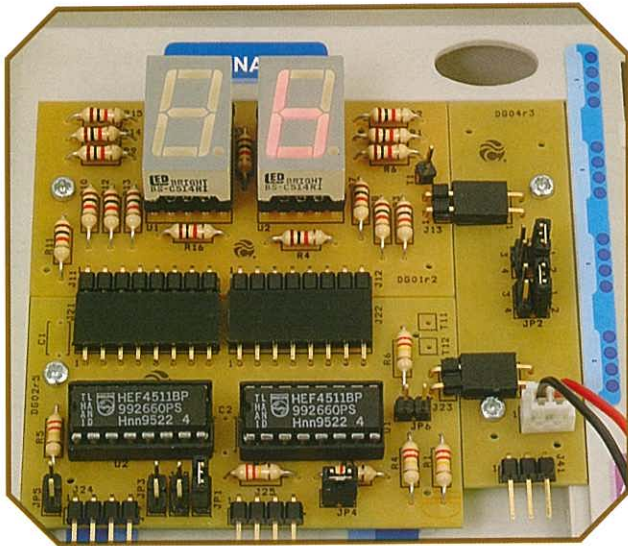
trebbe essere illuminato in modo tenue e instabile, in quanto gli ingressi del driver sono in aria è quindi non hanno un livello logico fisso.

La prima prova consiste nell'inserire il ponticello JP5 del circuito DG02, il resto dei ponticelli di questo circuito non deve essere montato. Collegando questo ponticello si devono illuminare i sette segmenti di ogni display, se qualcuno di essi non si illumina è probabile che ci sia qualche problema di collegamento, sulle saldature, sulle connessioni tra il display e lo zoccolo, oppure tra qualcuno degli integrati e il proprio zoccolo. Dopo aver realizzato questa prova si toglie il ponticello da JP5.

Montando JP6 si impedisce la visualizzazione.

Il ponticello per lo spegnimento totale è JP6, JP5 non deve essere inserito.

Spegnimento forzato

Partiamo dalla prova precedente dove abbiamo verificato che tutti i segmenti si illuminavano. Inserendo il ponticello su JP6 si devono spegnere completamente e non devono assolutamente lampeggiare. Al termine della prova bisogna estrarre il ponticello da JP6.

Prove con codici

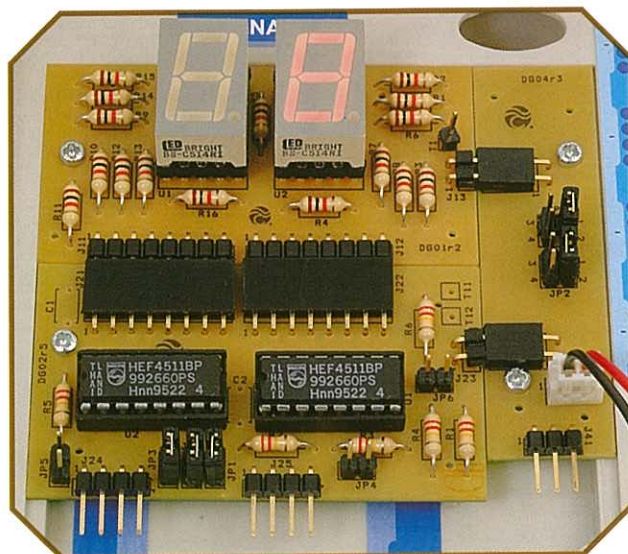
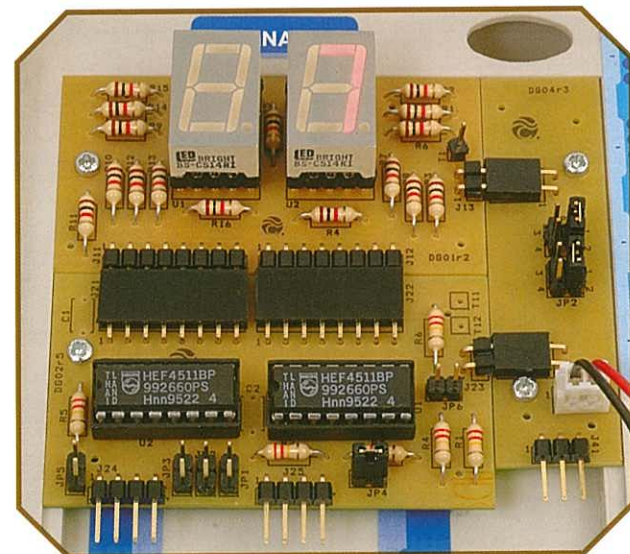
Benché questo circuito disponga di connettori per i codici, permette anche di inserirli in modo manuale per uno solo dei display, a questo scopo dispone di quattro ponticelli siglati

Se si tolgono JP5 e JP6, i ponticelli da JP1 a JP4 permettono di applicare un codice al display delle unità.

Prova con ponticelli su JP1 e JP4, si visualizza il 6.

Ponticelli su JP1, JP3 e JP4 per ottenere il 2.

da JP1 a JP4, dove JP1 è il bit meno significativo. Quando nessuno di questi ponticelli è installato il codice è 1111, questo valore è interpretato dal driver come errato perché corrisponde alla cifra decimale 15 e il driver accetta solamente dallo 0 al 9, quindi nessun segmento si illumina.

Iniziamo l'esperimento, per prima cosa verifichiamo che i ponticelli JP5 e JP6 siano montati. Inseriamo quindi i ponticelli su JP1 e JP4, il codice quindi sarà 0110 e si deve visualizzare il 6. Ogni volta che si collega un ponticello si applica uno 0, quando non c'è il ponticello la resistenza imposta il valore 1 per default.

Possiamo ripetere fino a 10 combinazioni possibili per ottenere differenti letture, vi sono anche delle combinazioni che non sono possibili, per queste non si deve illuminare alcun segmento e sono quelle che corrispondono ai valori decimali dal 10 al 15.

Vediamo qualche esempio in più. Se montiamo i ponticelli su JP1, JP3 e JP4 si deve visualizzare il 2, e se li montiamo su JP1, JP2 e JP3 il codice sarà 1000 e verrà visualizzato il numero 8.

Ponticelli su JP1, JP2 e JP3, si visualizza il numero 8.

Ponticello su JP4, si visualizza il 7.

La porta B

Un altro dei dispositivi del PIC 16F870 è la porta B, che ha 8 linee bidirezionali. L'indirizzo dei dati si indica sul registro TRISB. Se impostiamo a 1 un bit del registro TRISB il suo corrispondente bit del registro PORTB rimarrà configurato come ingresso. Se disattiviamo, impostandolo a 0, un bit del registro TRISB, il corrispondente bit del registro PORTB rimarrà configurato come uscita. Fino a ora l'unica differenza tra la porta A e la porta B consiste nel numero dei pin, però se analizziamo le funzioni e l'architettura interna di ognuna di queste porte potremo osservare grandi differenze.

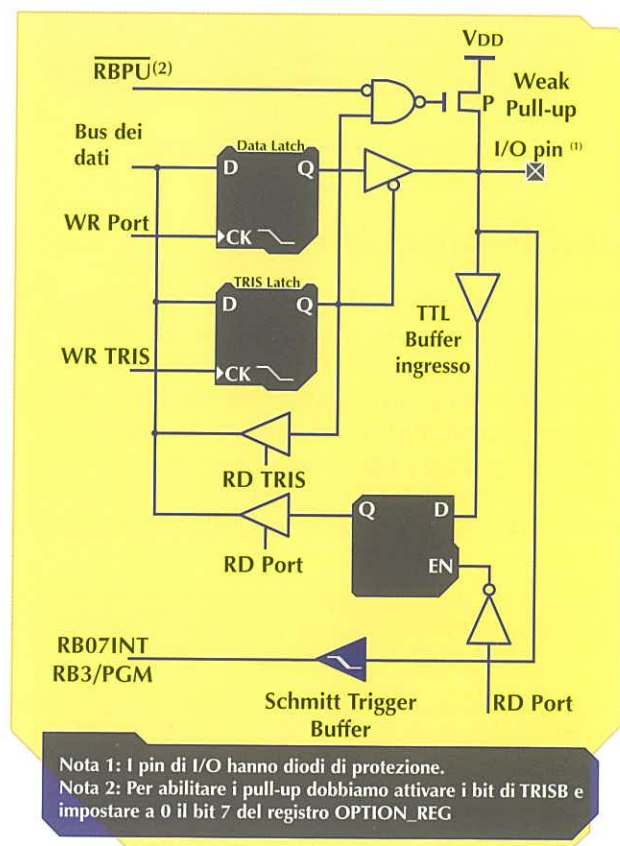
Funzioni dei terminali della porta B

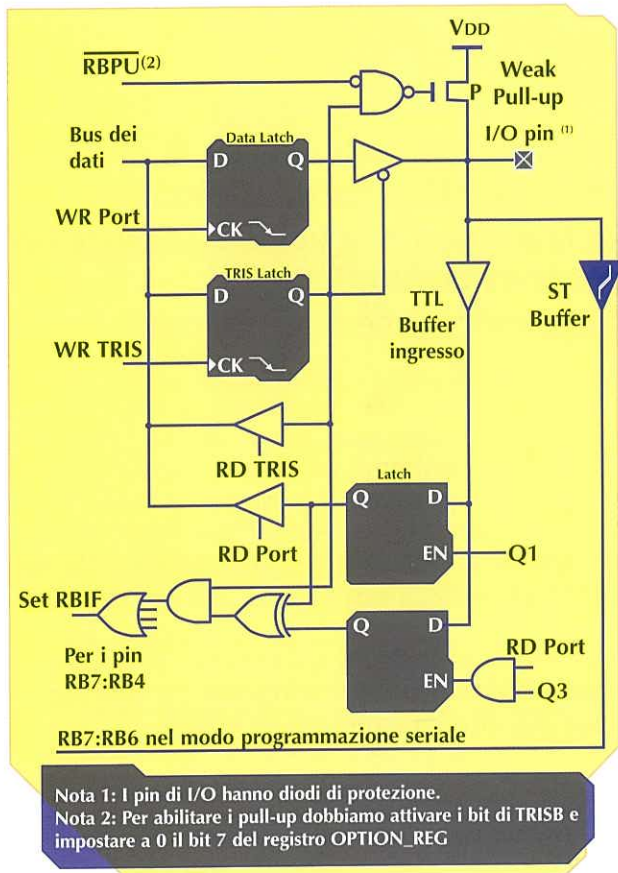
Come abbiamo già detto parlando della porta A, le caratteristiche dimensionali dei microcontroller impongono che la maggioranza dei loro pin possa realizzare più di una funzione. Come nella porta B, oltre a I/O digitali, alcuni pin hanno multiplexate altre funzioni.

Il terminale RB0 si può configurare come un ingresso/uscita digitale o come ingresso per un interrupt esterno. In questo secondo modo, attivando il terminale, il processore genera un interrupt che ferma l'esecuzione del programma in corso e risolve la routine di interrupt, che sono altre istruzioni il cui indirizzo è indicato sul vector di interrupt posto all'indirizzo 0004h della memoria di codice.

Studiando le caratteristiche del PIC abbiamo visto che ha la possibilità di essere programmato per via elettrica, oltre a permettere la programmazione a bassa tensione (5 volt). Se vogliamo utilizzare questa caratteristica applicheremo il positivo dell'alimentazione al pin 1 del PIC ($\overline{MCLR}$/Vpp) e la massa al pin RB3/PGM. La scrittura del codice sulla memoria si realizza sempre per via seriale: si applicano gli impulsi del clock per il sincronismo tra l'ingresso dei dati e il processore sul terminale RB6/PGC, e i dati vengono caricati via seriale tramite il terminale RB7/PGD.

Nome	Bit #	Buffer	Funzione
RB0/INT	bit 0	TTL/ST[1]	Ingresso/Uscita digitale o richiesta esterna di interrupt
RB1	bit 1	TTL	Ingresso/Uscita digitale
RB2	bit 2	TTL	Ingresso/Uscita digitale
RB3/PGM	bit 3	TTL/ST[1]	Ingresso/Uscita digitale o massa per la programmazione a bassa tensione
RB4	bit 4	TTL	Ingresso/Uscita digitale
RB5	bit 5	TTL	Ingresso/Uscita digitale
RB6/PGC	bit 6	TTL/ST[2]	Ingresso/Uscita digitale o clock nella programmazione seriale
RB7/PGD	bit 7	TTL/ST[2]	Ingresso/Uscita digitale o dati nella programmazione seriale

(1) Questo buffer avrà un ingresso tipo ST (Schmitt Trigger) quando sarà programmato come interrupt o programmazione in stato basso (bit 0 e bit 3).
(2) Questo buffer sarà un ingresso tipo ST quando si utilizza per programmazione seriale.

Nome e funzione dei terminali della porta B.

Architettura interna dei terminali RB <3:0>.

Architettura interna dei terminali RB <7:4>.

Architettura interna

Tutti i terminali della porta B dispongono di un pull-up interno. Mediante il bit $\overline{\text{RBPU}}$, bit più significativo del registro OPTION_REG, i terminali della porta possono essere collegati al positivo dell'alimentazione (impostando a 0 $\overline{\text{RBPU}}$) grazie al pull-up. Quando i terminali della porta sono configurati come uscita vengono automaticamente scollegati dal pull-up.

Se si genera un reset per collegamento dell'alimentazione (POR) si scollegano tutti i pull-up.

Osservando la figura possiamo vedere che il pull-up è un transistor CMOS di tipo P, quindi per l'abilitazione del pull-up è necessario che RBPU sia disattivato (a 0) e che il latch di TRISB contenga un 1 (ingresso).

Quattro terminali della porta B, RB <7:4>, possono essere programmati per generare un interrupt quando cambia il loro stato logico. Questo interrupt si potrà generare solamente se i terminali saranno configurati come ingressi. I terminali di ingresso RB7:RB4 si comparano con il valore acquisito (mantenuto sul latch) dall'ultima lettura di PORTB e se non coincidono i valori e il bit di abilitazione lo autorizza, verrà eseguita una richiesta di interrupt al processore. Quando si produce un interrupt di questo tipo, il bit RBIF del registro INTCON (INTCON<0>) si attiva (passa a 1) automaticamente. L'interrupt può risvegliare il microcontroller da uno stato di SLEEP.

L'utente, nella routine di servizio all'interrupt, può disattivare questo flag nei seguenti modi:

— Eseguendo una lettura o una scrittura di PORTB. Terminerà la condizione della comparazione.

— Disattivando il bit RBIF.

Questa caratteristica è molto utilizzata quando si lavora con una tastiera e vogliamo rilevare se è stato premuto un tasto. Premendo il tasto cambia lo stato dell'ingresso e si produce un interrupt che richiama la routine specifica per individuare il tasto premuto.

Per configurare il terminale RB0/INT per lavorare con un interrupt esterno, dobbiamo utilizzare il bit INTEDG del registro OPTION_REG (OPTION_REG <6>).

Dopo aver visto la PORTB, o porta B, dato

Indirizzo	Nome	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valore in POR o in BOR	Valore negli altri reset
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	Configurazione dei pin della porta B								11111111	1111 1111
81h, 181h	OPTION_REG	RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	11111111	1111 1111

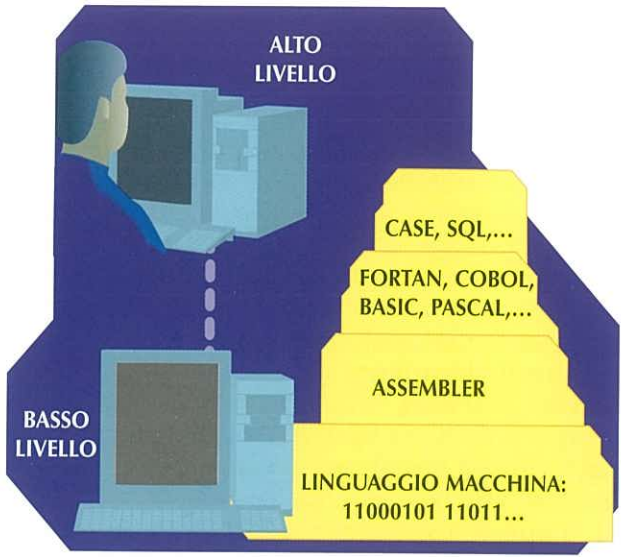
x: sconosciuto, u: inalterato; le celle ombreggiate non intervengono nella configurazione della porta B.
Registri associati alla porta A.

clrf	PORTB	Cancella i latch di uscita
bsf	STATUS,RP0	Seleziona banco 1
movlw	b'11111111'	
movwf	TRISB	La porta B si configura come ingressi digitali

Esempio di configurazione della porta B.

che si utilizzano indistintamente entrambe le denominazioni, chiariamo una delle idee fondamentali per ottenere una buona programmazione. Quando la complessità di un programma è alta, è necessario essere molto ordinati per evitare di creare una situazione confusa, dobbiamo ricordare che i grandi programmi esigono il lavoro perfettamente organizzato di molti programmatori.

Metodologia

Con la varietà di linguaggi di programmazione esistente e le caratteristiche particolari di ogni progettista, se non si rispettano le regole diventa molto difficile interpretare i programmi. Le stesse regole si possono applicare alla maggioranza dei linguaggi di programmazione e praticamente ai progettisti di tutto il mondo. Così, se dovremo mettere mano a un programma che non abbiamo sviluppato noi, saremo capaci di interpretarlo e anche di modificarlo.

Classificazione dei linguaggi di programmazione.

Si chiama "Metodologia della Programmazione" l'insieme di regole che si seguono nella fase di progetto del programma.

È fondamentale strutturare il codice di un'applicazione e applicare delle regole. Nelle figure possiamo osservare la differenza esistente tra un codice che non segue nessuna regola di programmazione e lo stesso codice strutturato. Supponiamo ora un programma molto più esteso che venga realizzato senza seguire alcuna regola, risulterà un caos anche per lo stesso programmatore che lo avrà creato.

```
;
; Il Display a 7 segmenti
; Sul disply a catodo comune collegato alla porta B, visualizziamo lo stato
; logico "0" o "1" dell'interruttore RC0. Mediante l'interruttore RC1 si attiva o meno il
; punto decimale.

List    p=16F870;Tipo di processore
include "p16F870.INC";Definizione dei registri interni
Org 0X00
Inizio  clrf    PORTB;Cancella i valori residui
bsf     STATUS, RP0;Seleziona banco 1
clrf TRISB;Porta B si configura come uscita
movlw   0XFF
movwf   TRISC
bcf STATUS, RP0;Seleziona banco 0
Lopp    clrwdt  ;Aggiorna il WDT
btfsc   PORTC, 0;Test su RC0
goto    RC0_è_1;È a livello "1"
movlw   b'00111111'
movwf   PORTB   ;Visualizza il numero 0
goto    Test_RC1
RC0_è_1 movlw   b'00000110'
movwf   PORTB   ;Visulizza il numero 1
Test_RC1 btfsc  PORTC,1;Test su RC1
goto    RC1_è_1;È a "1"
bcf     PORTB, 7;Disconnetti punto decimale
goto Loop
RC1_è_1:        bsf     PORTB,7         ;Attiva punto decimale
goto Loop
end
```

Esempio di programma senza seguire le regole di programmazione.

```
; Il Display a 7 segmenti

; Sul disply a catodo comune collegato alla porta B, visualizziamo lo stato
; logico "0" o "1" dell'interruttore RC0. Mediante l'interruttore RC1 si attiva o meno il
; punto decimale.

            List    p=16F870            ;Tipo di processore
            include "p16F870.INC"       ;Definizione dei registri interni

            Org 0X00

Inizio      clrf    PORTB               ;Cancella i valori residui
            bsf     STATUS, RP0         ;Seleziona banco 1
            clrf    TRISB               ;Porta B si configura come uscita
            movlw   0XFF
            movwf   TRISC
            bcf     STATUS, RP0         ;Seleziona banco 0

Lopp        clrwdt                      ;Aggiorna il WDT
            btfsc   PORTC, 0            ;Test su RC0
            goto    RC0_è_1             ;È a livello "1"
            movlw   b'00111111'
            movwf   PORTB               ;Visualizza il numero 0
            goto    Test_RC1

RC0_è_1     movlw   b'00000110'
            movwf   PORTB               ;Visulizza il numero 1

Test_RC1    btfsc   PORTC,1             ;Test su RC1
            goto    RC1_è_1             ;È a "1"
            bcf     PORTB, 7            ;Disconnetti punto decimale

            goto Loop
RC1_è_1:    bsf     PORTB,7             ;Attiva punto decimale
            goto Loop
```

Lo stesso esempio di programma, però strutturato.

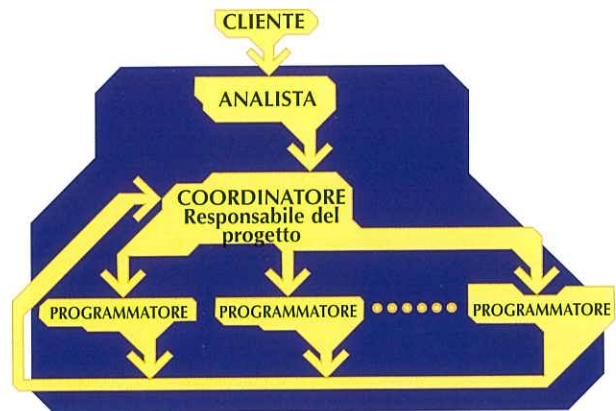
Linguaggi di programmazione

Il linguaggio di programmazione è il mezzo col quale comunichiamo con il computer e comprende l'insieme di note, regole, simboli, espressioni, ecc. che descrivono algoritmi e strutture dei dati. Benché esistano moltissimi concetti comuni a tutti i linguaggi, la grande varietà di applicazioni fa sì che esistano molti linguaggi di programmazione differenti e che vengano applicante anche diverse metodologie.

Esistono linguaggi specifici per il calcolo numerico (Fortran, Matlab), gestione (Cobol), programmazione Web (HTML), orientati a oggetti (Visual Basic), per il real time (C, Assembler), di utilizzo generale (Pascal, C), ecc. Alcuni di questi linguaggi si adattano a quello del proprio programmatore e in seguito traducono il codice creato in uno leggibile per il computer, sono chiamati linguaggi di alto livello; esistono anche linguaggi di basso livello i cui algoritmi sono vicini al codice macchina ed è il programmatore a doversi adattare a essi per sviluppare il codice.

I linguaggi di basso livello utilizzano un compilatore per tradurre le istruzioni in codice binario della macchina. I linguaggi di alto livello utilizzano un interprete che traduce il codice creando istruzioni di basso livello che successivamente, a loro volta, verranno trasformate in binario.

Per programmare il nostro microcontroller utilizzeremo l'assembler. Il linguaggio assembler è un linguaggio di basso livello che fornisce l'opportunità di conoscere a fondo le operazioni del processore, mantenendo uno stretto controllo su di esse. Anche se l'utilizzo dell'assembler richiede l'impiego di un maggior numero di istruzioni per eseguire un'azione, dota i programmi di maggior rapidità e li rende più compatti, ottimizzando la loro esecuzione.

Evoluzione storica dei linguaggi di alto livello.

Struttura tipica di sviluppo di un progetto.

Organizzazione

Ora conosciamo l'importanza di utilizzare una metodologia al momento di programmare, però cosa succede se sviluppiamo il nostro lavoro di programmatori per un'azienda?

Molti dei progetti che si realizzano in un'azienda hanno un grado di complessità elevato. In questi casi il progetto viene affrontato congiuntamente da un gruppo di programmatori.

Quindi gli analisti avranno l'incarico di specificare cosa deve fare il nostro programma. Essi definiranno il progetto nel suo insieme indicando i modelli dello sviluppo.

Ci sarà anche un altro coordinatore con il compito di assegnare a ogni programmatore la parte da sviluppare e, solitamente, di assemblare tutte le parti in un insieme globale. Ogni programmatore svilupperà un compito specifico.

È molto importante che tutti seguano le stesse regole di programmazione: identificazione dell'autore, utilizzo di variabili, tabulati, commenti, utilizzi di maiuscole e minuscole, ecc., sono alcuni degli aspetti che si raccolgono nei capitolati tecnici di un'azienda.